

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»**  
**ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ**  
(повна назва інституту/факультету)

**КАФЕДРА БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ**  
(повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри

О. В. Лебедєв  
(підпис) (ініціали, прізвище)

“ ” \_\_\_\_\_ 2019 р.

**Дипломна робота**  
на здобуття ступеня бакалавра

з напрямку підготовки 6.051402 «Біомедична інженерія»  
(код та назва)

на тему: «Програмний комплекс для визначення професії на основі тесту Амтхауера»

Виконав : студент 4 курсу, групи БМ-51  
(шифр групи)

Луценко Катерина Вікторівна  
(прізвище, ім'я, по батькові) (підпис)

Керівник ст. викл. каф. БМІ, Білошицька О. К.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент доц. каф. БЗЛ, к.м.н., доцент Худецький І.Ю.  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Консультант 3 доц.каф. ОППЦБ, к.т.н., доц. Демчук Г.В.  
(номер розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Нормоконтроль інженер 1 категорії Андреев П. І.  
(посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет

Кафедра

Рівень вищої освіти

Напрямок підготовки

Біомедичної інженерії

Біомедичної інженерії

Перший (бакалаврський)

6.051402 «Біомедична інженерія»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

О. В. Лебедєв  
(підпис) (ініціали, прізвище)

“    ”                      2019 р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

Луценко Катерині Вікторівні

(прізвище, ім'я, по батькові)

1. Тема роботи Програмний комплекс для визначення професії на основі тесту Амтхауера.

Керівник роботи Білошицька Оксана Костянтинівна ст.викл. каф. БМІ,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 27 » травня 2019 р. № 1404 - с

2. Строк подання студентом роботи 10.06.2019

3. Вихідні дані до роботи 9 субтестів тесту структури інтелекту Амхауера.

4. Зміст дипломної роботи: провести огляд літератури, яка стосується тесту структури інтелекту Амтхауера; проаналізувати існуючі мови програмування; розробити алгоритм роботи програми; розробити програму для проходження тесту; налаштувати передачу даних в один файл.

5. Перелік ілюстративного матеріалу презентація дипломної роботи.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
3	Демчук Г.В., доцент кафедри охорони праці, промисловості та цивільної безпеки		

7. Дата видачі завдання \_\_\_\_\_

## Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел.		
2	Аналіз мов програмування.		
3	Вибір середовища програмування для створення програмного комплексу.		
4	Реалізація програмного комплексу		
5	Налаштування передачі даних в один файл.		
6	Оформлення ДР.		
7	Підготовка до захисту ДР.		
8	Захист ДР.		

Студент

\_\_\_\_\_  
(підпис)

К.В. Луценко

(ініціали, прізвище)

Керівник роботи

\_\_\_\_\_  
(підпис)

О.К. Білошицька

(ініціали, прізвище)

## АНОТАЦІЯ

Тема дипломної роботи практики: «Програмний комплекс для визначення професії на основі тесту Амтхауера».

Обсяг пояснювальної записки становить 60 сторінок, міститься 15 ілюстрацій, 15 таблиць. Загалом опрацьовано 30 джерел.

Актуальність теми обумовлена тим, що дослідження аналітичного центру НУЦД показало, що лише близько третини підлітків (31%) у віці від 14 до 17 років твердо визначилися з вибором майбутньої професії. Більшість випускників цього та наступних років розглядають кілька варіантів професійного розвитку – таких 54%. Не визначилися з вибором 15% опитаних.

Метою дипломної роботи є розроблення та кодування тесту Амтхауера з подальшим збереженням отриманих даних у налаштований для цього комп'ютер в один файл.

Для досягнення мети були поставлені наступні задачі:

1. Розглянути особливості існуючих програмних та веб-інтерфейсних засобів для проходження тесту Амтхауера.
2. Обрати середовище програмування.
3. Створити програму для проходження тесту Амтхауера у обраному середовищі з передачею даних по локальній мережі в один файл.

Основні результати: робота виконана для впровадження в навчальний процес, проведено огляд літературних джерел, що стосуються структури та надійності тесту Амтхауера, обрано мову програмування та налаштовано передачу даних в один файл, створено графічний інтерфейс програми.

Ключові слова: тест, тест Амтхауера, професія, Java, локальна мережа.

## SUMMARY

The theme of diploma work is «Software package based on Amthuera test for profession determination».

The volume of the explanatory note is 60 pages, contains 15 illustrations, 15 tables. In total, 30 sources have been processed.

Actuality is the urgency of the topic is due to the fact that the research of the analytical center NUCE showed that only about one third of adolescents (31%) aged 14 to 17 years firmly decided on the choice of future profession. Most graduates of this and subsequent years are considering several options for professional development - such 54%. Not decided with the choice of 15% of the respondents.

Aim is the developing and encoding the Amthuere test, then saving the data in a configured computer for that file.

To achieve the goal, the following tasks were set:

1. Consider the features of existing software and web interface tools for passing the test of Amthauer.
2. Select the programming environment.
3. Create a program for passing Amthuera test in a selected medium with the transfer of data on the local network into one file. Main results: reviewed literature on the structure and reliability of the test of Amtahuaaur, the language of programming was selected and the data transmission was configured in one file, the graphical interface of the program was created.

Keywords: test, Amthuera test, profession, Java, local network.

## ЗМІСТ

СПИСОК УМОВНИХ СКОРОЧЕНЬ .....	8
ВСТУП .....	9
РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРИ .....	10
1.1 Сутність тесту Амтхауера .....	10
1.2 Структура тесту Амтхауера.....	11
1.3 Опис субтестів .....	12
1.4 Інтерпретація результатів тесту .....	13
1.5 Надійність тесту .....	14
1.5 Оновлена версія тесту .....	15
Висновок до розділу 1 .....	20
РОЗДІЛ 2 .....	22
2.1 Вибір мови програмування.....	22
2.2 Переваги Java як мови програмування для віконної програми.....	24
2.3 Створення інтерфейсу.....	28
РОЗДІЛ 3 .....	36
РЕАЛІЗАЦІЯ ТЕСТУ АМТХАУЕРА.....	36
3.1 Блок-схема алгоритму.....	36
3.2 Результат програмування.....	40
Висновок до розділу 3 .....	43

					БМ51.08.2705.1404.ПЗ			
Вим.	Лист	№ докум.	Підпис	Дата				
Розробив		Луценко К.В.			Програмний комплекс для визначення професії на основі тесту Амтхауера	Літ.	Лист	Листів
Перевірів		Білошицька О. К.						
Реценз.		Худецький І. Ю..				КПІ ім. І. Сікорського ФБМІ БМ-51		
Н. Контр.		Андрєєв П. І.						
Затвердив		Лебедев О.В.						

РОЗДІЛ 4.....	44
4.1. Загальна характеристика умов праці у навчальну приміщені.....	44
4.2 Відповідність вимогам об'єму і площі на одного працівника та розташування технологічного обладнання .....	46
4.3 Оцінка небезпечних і шкідливих виробничих факторів .....	47
4.4 Шум .....	47
4.5 Електронебезпека .....	48
4.6 Пожежна безпека.....	50
Висновок до розділу 4.....	51
ВИСНОВКИ .....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	53
ДОДАТОК А.....	57
ДОДАТОК Б .....	60

## СПИСОК УМОВНИХ СКОРОЧЕНЬ

- ГІТ – груповий інтелектуальний тест  
ООП – об’єктно-орієнтоване програмування  
Тест – тест структури інтелекту Амтхауера  
GUI – графічний інтерфейс користувача  
SWT – Standart Widget Tool

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
Изм.	Лист	№ докум.	Підпис	Дата		8



## ВСТУП

Темою моєї дипломної роботи є «Програмний комплекс для визначення професії на основі тесту Амтхауера».

Актуальність теми обумовлена тим, що дослідження аналітичного центру НУЦД показало, що лише близько третини підлітків (31%) у віці від 14 до 17 років твердо визначилися з вибором майбутньої професії. Більшість випускників цього та наступних років розглядають кілька варіантів професійного розвитку – таких 54%. Не визначилися з вибором 15% опитаних.

Тест Амтхауера – тест на визначення профорієнтації, має високі оцінки достовірності результатів, тому створення програмного комплексу на основі цього тесту є доцільним.

Наразі існують тільки друковані та онлайн-версії тесту Амтхауера, що не дає змогу впровадити даний тест у навчальний процес. Програмування даного тесту та подальше налаштування збереження отриманих даних від групи студентів в один файл є актуальним, оскільки, окрім появи комп'ютерної версії тесту, викладач або куратор групи студентів отримає файл з відповідями протестованих студентів, що пришвидшить аналіз тесту та спростить його проведення у великої групи людей.

Метою даної роботи є створення програмного комплексу для визначення здібностей особистості на основі тесту Амтхауера.

Для досягнення запланованої мети дипломної роботи були поставлені наступні задачі:

1. Розглянути особливості існуючих програмних та веб-інтерфейсних засобів для проходження тесту Амтхауера.
2. Обрати середовище програмування.
3. Створити програму для проходження тесту Амтхауера у обраному середовищі з передачею даних по локальній мережі в один файл.

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						9
Изм.	Лист	№ докум.	Підпис	Дата		

## РОЗДІЛ 1

### ОГЛЯД ЛІТЕРАТУРИ

#### 1.1 Сутність тесту Амтхауера

Рудольф Амтхауер розглядав інтелект як спеціалізовану підструктуру в цілісній структурі особистості, тісно пов'язану з її емоційно-вольовою сферою, потребами і ціннісними установками. В його уяві інтелект – це сформована при житті система розумових операцій, яка «вписана» в особистість і відповідає її особливостям. Саме ж поняття «інтелект» є абстрактним, родовим терміном. Вивчення інтелекту повинно йти по шляху виділення типів і відповідних їм структур [1]. З позиції Р. Амтхауера однозначно випливає, що достовірність висновків про інтелектуальні здібності індивіда можлива за умови визначення їх типу, тобто на основі комплексного аналізу результатів за всіма субтестами і виділення домінуючого «центру ваги». Прогнози за результатами виконання окремих субтестів будуть менш надійні. Якщо для діагностики взагалі використовуються тільки окремі субтести, то про будь-яку достовірність висновків говорити не доводиться. Р. Амтхауер намагався створити цілісний тест структури інтелекту, а не набір субтестів, які можна використовувати на свій розсуд [2].

При розробці тесту автор дотримувався принципу найбільшої кореляції кожного субтесту із загальним результатом при найменшій інтеркореляції між субтестами. Частково це було досягнуто. Експериментальна апробація методики показує, що статистичний зв'язок показників субтестів із загальним результатом коливається від 0,8 ( $p \leq 0,01$ ) до 0,45 ( $p \leq 0,05$ ). Друга частина принципу конструювання тесту доведена не зовсім переконливо, так як кореляції субтестів між собою складають значення від 0,62 до 0,2 [3].

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						10
Изм.	Лист	№ докум.	Підпис	Дата		

Німецька стандартизація проводилась на вибірці з 8642 випробуваних. Після стандартизації нове середнє дорівнювало 100, стандартне відхилення дорівнювало 10.

## 1.2 Структура тесту Амтхауера

Тест складається з дев'яти субтестів, кожен з яких спрямований на вимірювання різних функцій інтелекту, у всіх субтестах (крім 4, 7 і 9) використовуються завдання закритого типу. Перед початком виконання кожного субтесту випробувані інструктуються. Їм пропонується виконати задачі подібного типу (тренувальні). Потім реєструється час виконання кожного субтесту, що складає від 6 до 10 хвилин. Загальний час тесту – 90 хвилин [3].

Тест Амтхауера містить такі субтести:

1. На загальну обізнаність і інформованість у різних областях знань (не тільки наукових, але і життєвих).
2. На класифікацію понять.
3. На встановлення аналогій.
4. На підведення двох понять під загальну категорію (узагальнення).
5. На вміння вирішувати прості арифметичні задачі.
6. На вміння знаходити числові закономірності.
7. На вміння подумки оперувати зображеннями фігур на площині.
8. На вміння подумки оперувати зображеннями об'ємних фігур.
9. На заучування слів.

Кожен субтест, за винятком четвертого, складається з 20 завдань, в четвертий субтест входять 16 завдань [4].

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						11
Изм.	Лист	№ докум.	Підпис	Дата		

### 1.3 Опис субтестів

1. Доповнення речень (логічний відбір) – дослідження індуктивного мислення, акцент на почуття реальності. Завдання випробуваного – закінчити речення одним з наведених слів. Кількість завдань – 20. Час виконання – 6 хв.

2. Визначення спільних рис (виключення зайвого слова) – дослідження здатності до абстрагування, оперування вербальними поняттями, почуття мови, індуктивне мовне мислення. У кожному завданні випробуваному пропонують п'ять слів, з яких чотири об'єднані смисловим зв'язком, а одне – зайве. Це слово і слід виділити у відповіді. Кількість завдань – 20, час їх виконання – 6 хв.

3. Аналогії – аналіз комбінаторних здібностей, рухливості мислення, розуміння відносин, задоволеності приблизними рішеннями. У кожному завданні випробуваному пропонуються три слова, між першим і другим існує певний зв'язок. Після третього слова – прочерк. З п'яти доданих до завдання варіантів відповіді необхідно вибрати таке слово, яке було б пов'язане з третім так само, як і перші два. Кількість завдань – 20, час виконання – 7 хв.

4. Класифікація (узагальнення) – оцінка здатності робити судження, здатності до абстракції, утворення понять, вміння словесно висловлювати думку. Випробуваний повинен позначити два слова загальним поняттям. Кількість завдань – 16, час виконання – 8 хв. Оцінка варіює від 0 до 2 балів в залежності від рівня узагальнення.

5. Завдання на підрахунок (арифметичні задачі) – оцінка рівня розвитку практичного математичного мислення. Субтест складається з 20 арифметичних задач. Час виконання – 10 хв.

6. Ряди чисел – теоретичне математичне мислення, аналіз індуктивного мислення, здатності оперувати числами, знаходити закономірності. У 20 завданнях необхідно встановити закономірність числового ряду і продовжити його. Час виконання – 10 хв.

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						12
Изм.	Лист	№ докум.	Підпис	Дата		

7. Вибір фігур (вибір геометричного зразка) – дослідження просторової уяви, комбінаторних здібностей, наочного цілісного мислення; приєднуються конструктивні компоненти. Випробуваному показують картки, де зображені розділені на частини геометричні фігури. При виборі відповіді слід знайти картку з фігурою, яка відповідає розділеній на частини. Кількість завдань – 20. Час виконання – 7 хв.

8. Завдання з кубиками – просторова уява; приєднуються аналітичні компоненти. У кожному з 20 завдань пред'являється куб в певному положенні, зміненому по відношенню до ряду кубів, позначених буквами. Необхідно ідентифікувати даний куб з одною з позначених букв. Час рішення – 9 хв.

9. Пам'ять. Завдання на здатність зосередити увагу і зберегти в пам'яті засвоєне. Випробуваний повинен запам'ятати ряд слів і знайти їх серед інших, пропонованих в завданні. Слова для запам'ятовування об'єднані в таблиці за певними категоріями. Всього пропонується запам'ятати 25 слів (час заучування таблиці – 3 хв.). Час виконання 20 завдань – 6 хв [3].

#### 1.4 Інтерпретація результатів тесту

При інтерпретації результатів тесту окремі субтести об'єднують в кілька комплексів:

1. Комплекс вербальних субтестів (субтести 1-4) передбачає загальну здатність оперувати словами як сигналами і символами. При високих результатах по цьому комплексу переважає вербальний інтелект, є загальна орієнтація на суспільні науки і вивчення іноземних мов. Практичне мислення є вербальним.

2. Комплекс математичних субтестів (5, 6) передбачає здібності в області практичної математики та програмування. Однаково високі результати по обом субтестам свідчать про «математичну обдарованість». Якщо ця обдарованість

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						13
Изм.	Лист	№ докум.	Підпис	Дата		

доповнюється високою результативністю по третьому комплексу, то, можливо, правильний вибір професії повинен бути пов'язаний з природно-технічними науками і відповідною практичною діяльністю [3].

3. Комплекс конструктивних субтестів (7, 8), що передбачає розвинені конструктивні здібності теоретичного і практичного плану. Однаково високі результати по субтестах цього комплексу є хорошою підставою не тільки для природно-технічної, але і загальнонаукової обдарованості. Якщо ж освіта не буде продовжена, то буде переважати прагнення до моделювання на рівні конкретного і наочного мислення, до вираженої практичної спрямованості інтелекту, до розвитку ручної вмілості і мануальних здібностей [4].

4. Комплекси теоретичних (2, 4) і практичних планів здібностей (1, 3). Результативність за цими тестами слід порівняти попарно, щоб більш точно сформулювати висновок про можливу професійну підготовку та успішність в навчанні.

### 1.5 Надійність тесту

#### 1. Загальний показник:

- а) методом розщеплення  $r = 0,969$ ;
- б) ретестова надійність через три дні  $r = 0,915$ , через три роки  $r = 0,748$ .

#### 2. Показники окремих субтестів:

- а) методом розщеплення  $r = 0,84-0,96$ ;
- б) ретестова надійність через три роки  $r = 0,15-0,68$ .

#### 3. Валідність тесту

Внутрішня валідність: середнє число кореляцій субтестів з загальним показником  $r = 0,60$ ; середнє число всіх інтеркореляцій субтестів  $r = 0,26$ .

#### 4. Кореляції загальної результативності з зовнішніми критеріями:

- а) оцінка інтелекту учня вчителем  $r = 0,62$ ;

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						14
Изм.	Лист	№ докум.	Підпис	Дата		

- б) успішність в школі  $r = 0,26-0,74$ ; в вузі  $r = 0,10-0,40$ ;  
 в) кореляції тесту з іншими тестами: тест Равена  $r = 0.35$ ; тест Векслера  $r = 0,62$  [5].

### 1.5 Оновлена версія тесту

У Німеччині в 2000 р була створена оновлена версія тесту (1-8-Т - 200011). Вона розроблена психологами Д. Ліпманом, Л. Бодуселом і Б. Броні спільно з Р. Амтхауера і суттєво відрізняється від колишніх редакцій цього тесту [2]. По-перше, в неї поряд з формами Л і В включена ще одна паралельна форма С, яка за своїми психометричеським параметрах повністю відповідає двом раннім формам. По-друге, субтести, що містилися в ранніх редакціях, були перероблені і доповнені завданнями на наукові знання з окремих їх циклам. По-третє, психологи відмовилися від примітивного і нічим не обґрунтованого виділення в результатах тесту теоретичного і практичного інтелектів. В рамках нової теоретично обґрунтованої і емпірично підтвердженої концепції структури інтелекту за допомогою версії IST-2000R можна виміряти наступні 11 показників: вербальний інтелект, просторово-фігурний інтелект, рахунковий інтелект, фігурне розуміння, умовиводи, вербальні знання, фігурні знання, математичні знання і загальні знання, а також флюїдний і кристалізований інтелект [3].

Таким чином, нова редакція тесту передбачає не тільки діагностику розумового розвитку за загальним тестового показником, але і більш глибокий аналіз, що спирається на сучасний так званий фасеточний (від англ. Facet - грань, сторона) підхід в діагностиці [4]. Суть цього підходу полягає в об'єднанні окремих показників (окремих психологічних процесів) для оцінки різних сторін, компонентів будь-якої характеристики (когнітивної або особистісної). Творці версії тесту 2000 при аналізі результатів застосували фасіткову модель, що й

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						15
Изм.	Лист	№ докум.	Підпис	Дата		

призвело до виділення ряду вищеописаних показників. Крім того, вони пропонують і аналіз параметрів кожного з субтестів, трактованих як вербальне, математичне та фігурне розуміння, вербальне і фігурне запам'ятовування і т.д.

Відзначимо, що вітчизняні психологи, які працювали з тестом Амтхауера, як впливає з опису запропонованих ними способів аналізу, ще на початку 1980-х рр., По суті, використовували подібну модель [5]. Крім того, вони ж, спираючись на концепцію соціально-психологічних нормативів К. М. Гуревича, розробляючи вітчизняні тести розумового розвитку ШТУР (Шкільний тест розумового розвитку), Астурія (Тест розумового розвитку для абітурієнтів та старшокласників) та ін. Першими використовували наукові знання (математичні, гуманітарні, природничі) в якості змісту тестових завдань. Такий підхід був використаний в версії тесту IST-2000R при конструюванні завдань на діагностику знань.

Існують два модулі тесту IST-2000R. Один – основний, або короткий, включає наступні три групи завдань, призначених для діагностики трьох видів інтелекту. Для оцінки вербального інтелекту використовуються три субтесту: «Завершення пропозицій», «Аналогії» і «Узагальнення». У субтесті «Завершення пропозицій» (Satzergänzung - SE) завдання складаються з пропозицій з відсутнім словом, яке слід вибрати з п'яти запропонованих так, щоб пропозиція стала правильним за змістом. У субтесті «Аналогії» (Analogieen - AN) в кожному завданні пропонуються три слова, між першими двома існує певний логічний зв'язок, а до третього слову потрібно підібрати з п'яти запропонованих на вибір таке, щоб логічний зв'язок між ними була аналогічною зв'язку першої пари. Завдання субтесту «Узагальнення» (Gemeinsamkeiten - GE) побудовані таким чином: у групі з шести слів слід вибрати два, які можна об'єднати загальним поняттям.

Для діагностики математичного інтелекту використовуються три субтести: «Арифметичні завдання», «Числові ряди» і «Математичні знаки». Субтест «Арифметичні завдання» (Rechenaufgaben - RE) перевіряє виконання

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						16
Изм.	Лист	№ докум.	Підпис	Дата		



простих арифметичних операцій з раціональними числами. Вони не представлені у вербальній формі, і тому вплив вербального розвитку виключається. У субтесті «Числові ряди» (Zahlenreihen - ZR) дані ряди чисел, побудовані за певними правилами. Потрібно для кожного ряду знайти наступне число, яке продовжує ряд. У субтесті «Математичні знаки» (Rechenzeichen - RZ) потрібно в кожному завданні вставити в рівняння один з чотирьох основних математичних знаків.

Фігурний інтелект оцінюється за допомогою наступних трьох субтестів: «Вибір фігур», «Задачі з кубиками» і «Матриці». У субтесті «Вибір фігур» (Figurenauswahl - FA) в кожному завданні дані п'ять частин, на які розрізані цілі фігури. Потрібно вибрати, яку з 10 запропонованих цілих фігур можна без залишку скласти з даних в завданні частин. У субтесті «Завдання з кубиками» (Wurfclaufgaben - WU) у верхньому рядку зображені кубики, три сторони яких добре видно. У кожному завданні представлений один з вище розташованих кубиків, але у зміненому положенні. Слід визначити, який з верхніх кубиків заданий в завданні. Кожне завдання субтесту «Матриці» (Matrizen - MA) являє собою композицію з пропущеним елементом. Потрібно вибрати із запропонованих елементів той, яким слід доповнити матрицю, щоб не порушити композицію. Автори тесту розглядають описані дев'ять субтестів як оцінюють розуміння, вміння думати, міркувати.

Кожен з дев'яти описаних субтестів включає 20 завдань.

Завдання на запам'ятовування складаються з двох субтестів: «Вербальне запам'ятовування» і «Фігурне запам'ятовування». В «вербальному запам'ятовуванні» потрібно запам'ятати запропоновані слова і поняття, що їх узагальнюють. На завершення випробовуваний повинен із запропонованих загальних понять (категорій) вибрати ті, до яких відносяться вивчені слова. У завданнях субтесту «Фігурне запам'ятовування» дано пари фігур. Після фази запам'ятовування пред'являється одна з фігур пари. Другу фігуру потрібно вибрати з п'яти альтернатив.

Час проведення основного модуля - 77 хвилин.

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
Изм.	Лист	№ докум.	Підпис	Дата		17

У розширений модуль тесту, крім перерахованих, входить тест на перевірку знань (Wissentest). Він складається з 46 питань, що відносяться до різних галузей знань (географія / історія, економіка, мистецтво / культура, математика, природничі науки і повсякденне життя / побут). Питання мають вербальну (16), числову (16) і образотворчу (14) форми. Випробуваний повинен вибрати правильну відповідь з п'яти варіантів.

Всього за допомогою розширеного модуля можна отримати 22 показника: дев'ять за окремим субтестом, сумарні показники вербального, математичного і фігурного інтелекту, загальний показник інтелекту, що об'єднує три види і є показником розуміння, умінь мислити; здатності до вербального та фігурного запам'ятовуванню, загальний показник запам'ятовування, сумуючий вербальний і фігурний показники; загальний (сумарний) показник за всіма попередніми показниками; характеристики знань вербальних, знань в числовій формі, знань в фігурній (образотворчої, наочної) формі, загальний показник знань (поінформованості); сумарний показник за всіма попередніми характеристикам.

Час проведення розширеного модуля - 130 хвилин.

Стандартизація тесту проведена на більш ніж 5800 випробуваних у віці від 13 до 60 років: учнів різних типів шкіл, гімназій, які отримують вищу і середню професійну освіту і кандидатів на отримання роботи. Для кожної з груп отримані статистичні норми (окремо для чоловіків і жінок): надійність-гомогенність, перевірена методом розщеплення, для окремих шкал - 0,88-0,96, ретестова надійність - 0,87-0,97.

Автори повідомляють про задовільну валідність тесту, отриманої як шляхом кореляції з показниками інших шести тестів інтелекту (у тому числі Тесту, вільного від впливу культури Р. Кеттелла і «Прогресивних матриць» Д. Равена), так і з оцінками успішності і успішністю професійної діяльності.

Крім такої перевірки валідності, Д. Лінман зі співавторами, використовуючи фасеточний підхід, знайшли значущі кореляції ( $r = 0,58$ ) флюїдного інтелекту, виміряного за допомогою IST-2000R, з показником тесту,

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
Изм.	Лист	№ докум.	Підпис	Дата		18

вільного від впливу культури, і кристалізованого інтелекту з тестом Д. Векслера для дорослих.

Ті ж дослідники розглядали взаємозв'язку показників інтелекту з 5-факторною моделлю особистості, виміряної за допомогою німецької версії NEO Five Factors Inventory [7]. Випробувані були 789 німців у віці 16 років - 21 роки. Ними знайдені негативні кореляції нейротизма (N) з флюїдним і кристалізувалися видами інтелекту, а також з вербальними і числовими факторами тесту інтелекту, на підставі чого був зроблений висновок про те, що нейротизм негативно впливає на виконання інтелектуальних завдань. Крім того, отримані значимі позитивні кореляції відкритий досвід (O) з вербальним і фігурним факторами, флюїдним інтелектом, виміряним по фасеточним моделям.

Таким чином, західні дослідники в даний час істотно модифікували відомий, добре зарекомендував себе тест інтелекту, розроблений Рудольфом Амтхауера, і використовують його не тільки в практичних цілях, але і для досліджень, спрямованих на оцінку співвідношення особистісного та когнітивного компонентів цілісної індивідуальності.

Груповий інтелектуальний тест (ГІТ) Дж. Вани (Vana Intelligence Test - VIT) розроблений словацьким психологом і відомий як надійний, валідний, добре зарекомендував себе в практиці школи інструмент. Тест містить сім субтестів:

- виконання інструкцій: спрямований на виявлення швидкості розуміння простих вказівок і їх здійснення, формально-динамічних характеристик протікання вербальних розумових процесів;
- арифметичні завдання: діагностує сформованого, математичних знань і дій, які засвоюються школярами в процесі навчання, вміння орієнтуватися в умовах завдання, розвиток математичного аналізу і синтезу;
- доповнення пропозицій: оцінює розуміння сенсу окремих пропозицій, розвиток мовних навичок, вміння оперувати граматичними структурами;

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						19
Изм.	Лист	№ докум.	Підпис	Дата		

- визначення подібності та відмінності понять: перевіряє вміння аналізувати поняття, порівнювати їх на основі виділення істотних ознак;
- числові ряди: виявляє вміння знаходити логічні закономірності побудови математичної інформації, аналітико-синтетичні математичні уміння;
- встановлення аналогій може: діагностує вміння мислити за аналогією, рівень розвитку операції узагальнення та аналізу логічних відносин;
- символи: перевіряє швидкісні можливості виконання простій розумової роботи, швидкість переробки невербальної інформації.

ГІТ розроблений у двох формах, які перевірені на взаємозамінність. На виконання кожного субтеста відводиться обмежений час (від 1,5 до 6 хвилин).

Адаптація тесту була проведена на учнів 3-х - 6-х класів міських та сільських шкіл (вибірка - більше 500 чоловік). Отримані результати свідчать про досить високої надійності та валідності цього тесту [8]. Крім загального бала по тесту, автори вітчизняної модифікації ГІТ пропонують способи оцінки гуманітарної та математичної спрямованості учнів, сформованість окремих вербально-логічних операцій, швидкісних характеристик розумової роботи і деякі інші лінії аналізу. ГІТ досить часто використовується для моніторингу розумового розвитку дітей 9-12 років у процесі навчання і високо оцінюється шкільними психологами як ефективна і прогностичність методика визначення як загального рівня, так і індивідуальних і групових особливостей розумового розвитку [9].

## Висновок до розділу 1

Вимоги до розумового розвитку підлітків, які містяться в тесті можна розглядати як модель нормативу, або, іншими словами, як умовний норматив. Такий шлях дає можливість відпрацювати багато методичних питань, пов'язаних із застосуванням нормативів в діагностиці.

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						20
Изм.	Лист	№ докум.	Підпис	Дата		

Перевагами та доцільністю використання тесту Амтхауера є такі факти:

1. Методика може проводитися як в індивідуальній, так і в груповій формі.
2. Можливість отримання інформації про розвиток «теоретичних» або «практичних» здібностей.
3. Поглиблене вивчення структурно-рівневих характеристик інтелекту і визначення типу вищої освіти, який індивід може освоїти.

Можливість визначення ступеня сформованості основних операцій мислення в поняттях, виділення сутнісного ознаки поняття (субтест 2); встановлення категоріальної приналежності, виділення класу (субтест 4); усвідомлення закономірних зв'язків між явищами (субтест 3).

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						21
Изм.	Лист	№ докум.	Підпис	Дата		

## РОЗДІЛ 2

### МЕТОДИ РЕАЛІЗАЦІЇ ТЕСТУ

#### 2.1 Вибір мови програмування

У таблиці 2.1.1 приведені найактуальніші вимоги до сучасних мов програмування.

Таблиця 2.1.1 Порівняння мов програмування по 10-бальній шкалі

	Швидкість обробки	Читабельність	GUI	Простота	2D графіка	Крос-платформеність
C	8	6	3	2	5	7
C++	8	6	4	3	6	7
C#	7	7	6	5	6	2
Java	6	7	7	6	7	10
Python	2	5	8	10	10	10
VB.net	6	10	10	8	5	2
Assembler	10	0	0	0	0	0

Коли потрібно якомога швидше розробити програмний продукт, вибір падає на високорівневі мови з максимально логічним і простим синтаксисом. Це, наприклад, Java. Кодування програми на таких мовах суттєво скорочує час розробки через велику кількість сторонніх бібліотек, максимального зрозумілого синтаксису. Швидкість виконання програм, написаних на даних мовах страждає, причому часом досить відчутно. Схема виконання на прикладі Java:



Рис. 2.1.1 – Схема виконання програми на мові Java

Найповільнішим блоком в даній схемі є аналізатор - він повинен байт-код програми транслювати «на льоту» в машинний код, при цьому витративши багато часу на точне визначення інструкції. Тому швидка розробка найчастіше означає повільне виконання.

Також програма повинна володіти потужною графічною частиною, що відповідає вимогам дизайну і юзабіліті. Розробка графічної частини часто вимагає досить багато часу, тому що відрізняється неабиякою складністю. Тут складність виникає в тому, що вивід графіки - це велика доля математики, а це означає, що є вимога до швидкості виконання, а через складність розробки є необхідність в високорівневій мові. В даному випадку дуже добре підходить C++/C#. Однак, якщо ГІК не дуже складний, але красивий, можливе використання Java, на якій створення красивих інтерфейсів набагато простіше, ніж на C++ і,

тим більше, С. Якщо програма орієнтована в першу чергу на «приховану роботу» з мінімумом взаємодії з користувачем, тоді вибір повинен лягати в бік швидких мов.

Можливість роботи програми на різних платформах, в різних операційних системах з мінімальними змінами забезпечує крос-платформеність. У цій сфері можна виділити такі мови: Java, C #, C ++ з різними бібліотеками.

Java створювався з тією умовою, що програми на даній мові повинні працювати на будь-якій платформі, де є JVM - Java Virtual Machine. Програми на Java взагалі не вимагають ніяких змін - після компіляції виходить .jar файл, який буде працювати і на Windows, і на Mac OS, і на Linux.

Оскільки, у розробці програмного комплексу для визначення професії на основі тесту Амтхауера найбільшу роль грають швидкість розробки, крос-платформеність та графічний інтерфейс користувача, було обрано мову програмування Java, яка найкраще підходить для віконного додатку, враховуючи вищезазначені фактори.

## 2.2 Переваги Java як мови програмування для віконної програми

Java включає в себе об'єктно-орієнтоване програмування (ООП) – концепцію, в якій можна не тільки визначати тип даних і його структуру, а й набір функцій, що застосовуються до нього. Таким чином, структура даних стає об'єктом, яким можна управляти для створення відносин між різними об'єктами.

При іншому підході - процедурному програмуванні - потрібно слідувати чітким інструкціям, використовувати змінні і функції. При ООП можна групувати ці змінні і функції за допомогою контексту, маркувати їх і посилатися на функції в контексті кожного конкретного об'єкта.

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						24
Изм.	Лист	№ докум.	Підпис	Дата		



Порівняння процедурного і об'єктно-орієнтованого програмування представлено на рисунку 2.2.1.

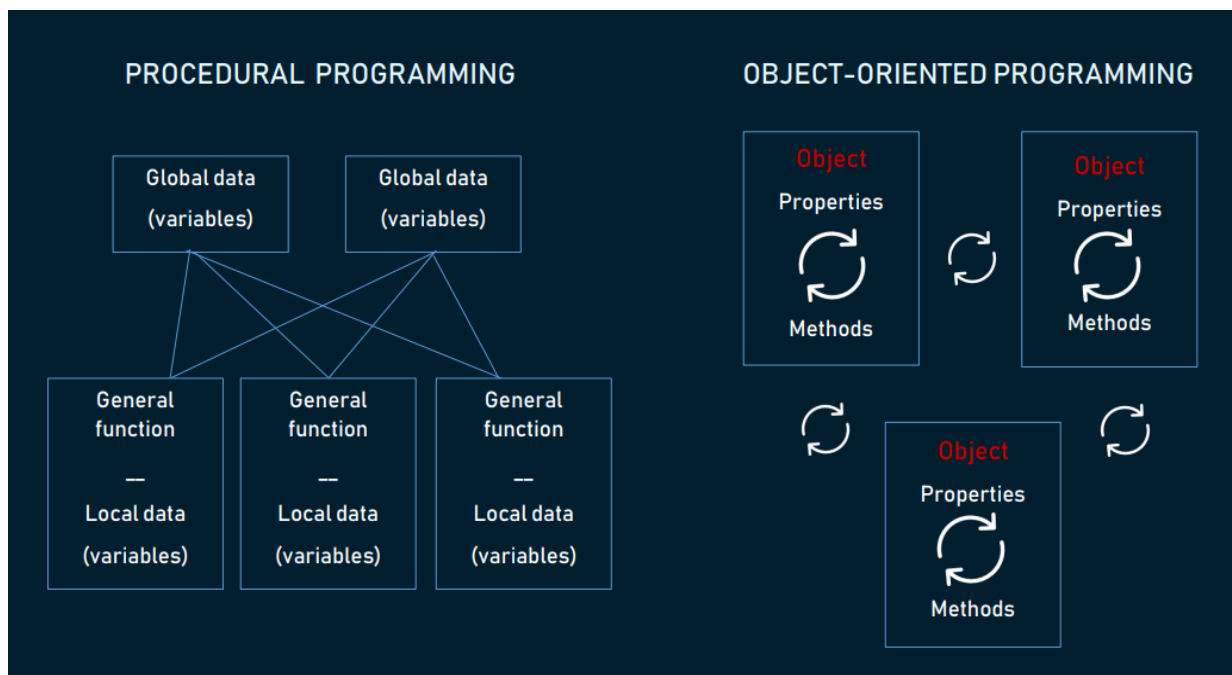


Рис.2.2.1 – Порівняння процедурного і об'єктно орієнтованого програмування

Java – мова високого рівня з простим синтаксисом і плавною кривою навчання. Java - це мова високого рівня, тобто вона схожа на людську мову. На відміну від мов низького рівня, які нагадують машинний код. Мови високого рівня перетворюються за допомогою компіляторів або інтерпретаторів. Це спрощує розробку, роблячи мову більш легкою для написання, читання і обслуговування.

Стандарт для корпоративних обчислювальних систем. Java підтримує безліч бібліотек - будівельних блоків будь-якої корпоративної системи. Бібліотеки допомагають розробникам створювати будь-які функції, які можуть знадобитися компанії. Java широко поширений - це мова, яку викладають в рамках введення в програмування в більшості шкіл і університетів. Можливості інтеграції Java вражають: більшість хостинг-провайдерів підтримують Java. Більш того, Java - мова, дешевий в обслуговуванні: працювати з Java можна з будь-якого комп'ютера, незалежно від конкретної апаратної інфраструктури.

Незалежність від платформи. Можна створити Java-додаток на Windows, скомпілювати його в байт-код і запустити його на будь-якій іншій платформі, яка підтримує віртуальну машину Java (JVM). Таким чином, JVM служить рівнем абстракції між кодом і обладнанням. Схема роботи Java показана на рисунку 2.2.2.

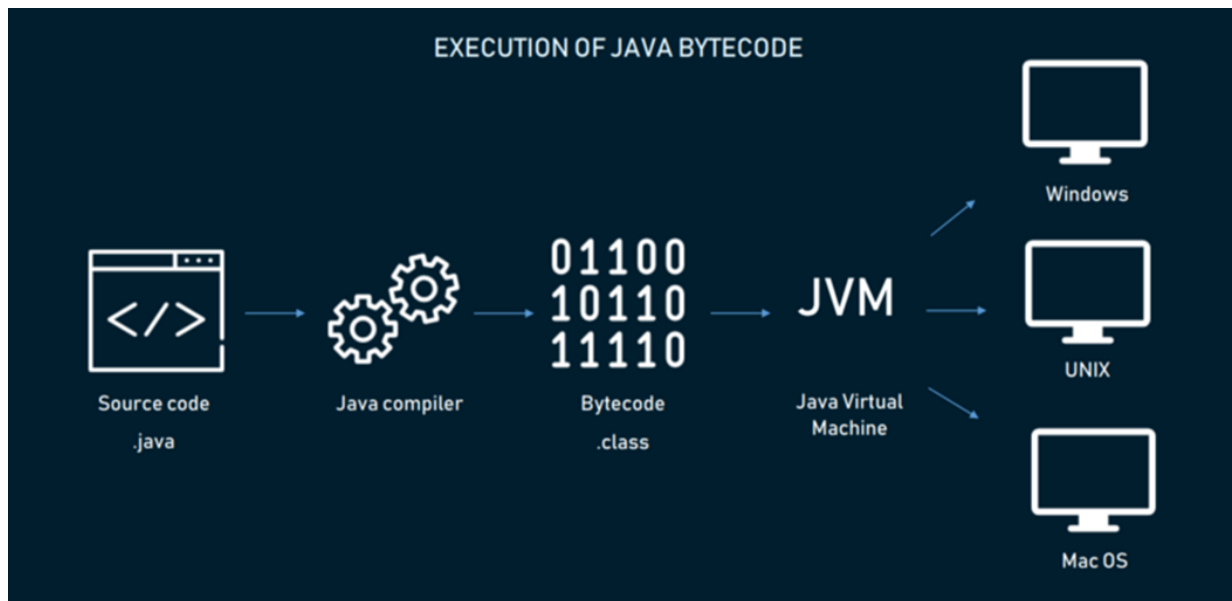


Рис. 2.2.2 – Схема роботи Java з різними платформами

Розподілене програмування. Java створювалася як мова для розподіленого програмування: він має вбудований механізм спільного використання даних і програм декількома комп'ютерами, що підвищує продуктивність праці. Порівняння процедурного і паралельного програмування представлено на рисунку 2.2.3.

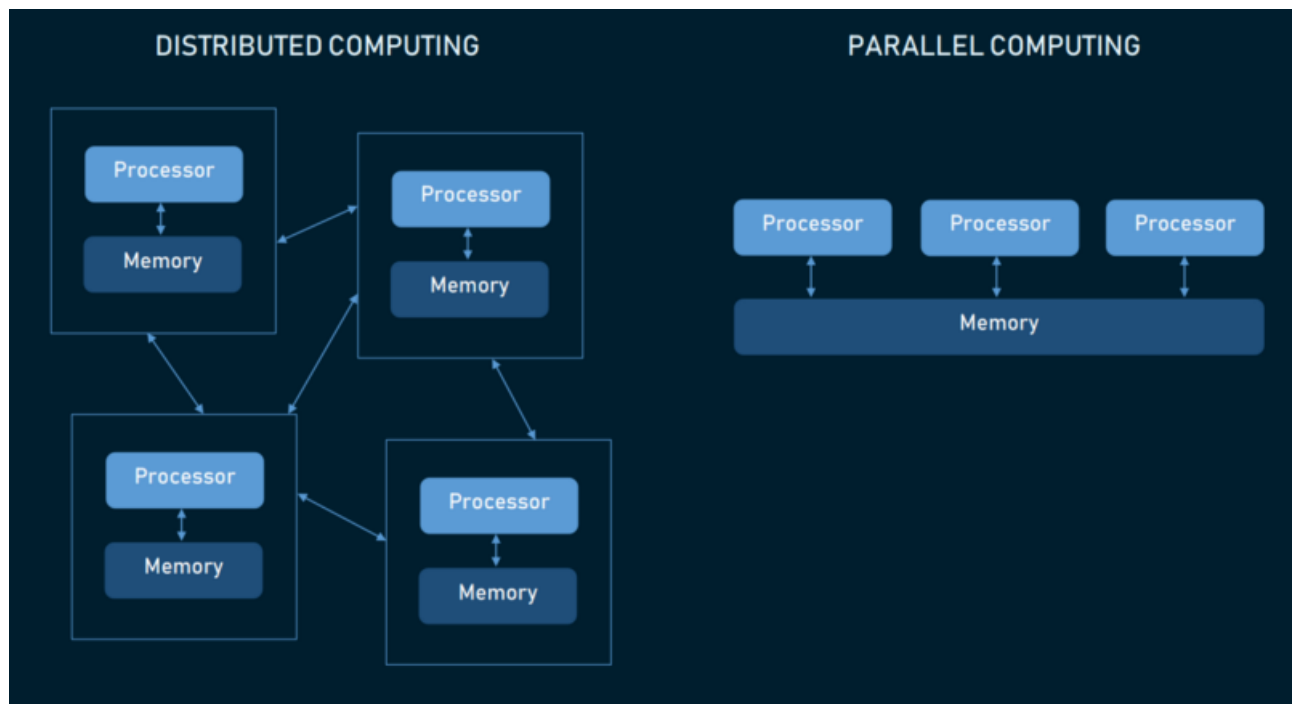


Рис. 2.2.3 – Порівняння паралельного і процедурного програмування

Багатопоточність. Потік - найменша одиниця обробки в програмуванні. Щоб максимально ефективно використовувати час процесора, Java дозволяє запускати потоки одночасно, що називається багатопоточність.

Потоки використовують одну і ту ж область пам'яті, тому між ними можна швидко перемикатися. Потоки незалежні один від одного: один потік не впливає на роботу інших потоків. Це особливо корисно в іграх і програмах з великим об'ємом анімації. Приклад багатопоточного виконання зображено на рисунку 2.2.4.



Рис. 2.2.4 – Приклад багатопоточного виконання

### 2.3 Створення інтерфейсу

Середовище розробки Eclipse дозволяє створювати різні Java додатки, а при використанні плагінів дочірніх проектів може служити відмінну IDE і для інших мов програмування. Однією з цікавих можливостей даного проекту є бібліотека побудови графічного інтерфейсу Standart Widget Toolkit (SWT). Це швидкодіюча альтернатива стандартним бібліотекам AWT і SWING, яка була розроблена підрозділом Rational фірми IBM і компанією Object Technology International (OTI). Дану бібліотеку можна успішно застосовувати для побудови призначеного для користувача інтерфейсу Java додатків [6-8].

Для створення інтерфейсу необхідно створити контейнер Java, в який будуть поміщені інші необхідні елементи графічного інтерфейсу. У цій дії контейнер буде створений за допомогою елемента JFrame. Контейнер буде поміщений в новий пакет, який буде відображатися в вузлі «Source Packages» [7].

У діалоговому вікні створення файлу вибираємо категорію Swing GUI Forms і тип файлу JFrame Form. Вводимо NumberAdditionUI як ім'я класу. Середовище IDE створює форму NumberAdditionUI і клас NumberAdditionUI в додатку NumberAddition і відкриває форму NumberAdditionUI в GUI Builder. Пакет my.NumberAddition замінює собою пакет за замовчуванням [8].

Для старту stand-alone SWT додатку потрібно додати в закладку «Libraries» властивостей проекту файл swt.jar, який знаходиться в каталозі plugins/org.eclipse.swt. <Платформа і версія> .ws. <Window system>. Шляхи до бібліотеки swt.jar для різних платформ наведені в таблиці 2.3.1.

Таблиця 2.3.1 – Налаштування шляху до бібліотеки swt.jar на різних платформах

ОС	Шлях до бібліотеки SWT
win32	INSTALLDIR\eclipse\plugins\org.eclipse.swt.win32_3.0.0\swt.jar
gtk	INSTALLDIR/eclipse/plugins/org.eclipse.swt.gtk_3.0.0/ws/gtk/swt.jar
motif	INSTALLDIR/eclipse/plugins/org.eclipse.swt.motif_3.0.0/ws/motif/swt.jar
photon	INSTALLDIR/eclipse/plugins/org.eclipse.swt.photon_3.0.0/ws/photon/swt.jar
macosx	INSTALLDIR/eclipse/plugins/org.eclipse.swt.carbon_3.0.0/ws/carbon/swt.jar

Для налагодження або запуску stand-alone SWT Java додатків в редакторі «VM arguments» потрібно вказати шлях до нативної бібліотеки SWT (зкладка «Arguments» панелі параметрів запуску програми). Варіанти налаштування шляхів для різних платформ наведені в таблиці 2.3.2 [7].

Таблиця 2.2 – Налаштування шляхів до нативної бібліотеки для різних платформ

ОС	Рядок параметрів
win32	-Djava.library.path=INSTALLDIRpluginsorg.eclipse.swt.win32_3.0.0/os/win32/x86
linux gtk	-Djava.library.path=INSTALLDIR/eclipse/plugins/org.eclipse.swt.gtk_3.0.0/os/linux/x86
linux motif	-Djava.library.path=INSTALLDIR/eclipse/plugins/org.eclipse.swt.motif_3.0.0/os/linux/x86
solaris motif	-Djava.library.path=INSTALLDIR/eclipse/plugins/org.eclipse.swt.motif_3.0.0/os/solaris/sparc
aix motif	-Djava.library.path=INSTALLDIR/eclipse/plugins/org.eclipse.swt.motif_3.0.0/os/aix/ppc
hpux motif	-Djava.library.path=INSTALLDIR/eclipse/plugins/org.eclipse.swt.motif_3.0.0/os/hpux/PA_RISC
photon qnx	-Djava.library.path=INSTALLDIR/eclipse/plugins/org.eclipse.swt.photon_3.0.0/os/qnx/x86
macosx	-Djava.library.path=INSTALLDIR/eclipse/plugins/org.eclipse.swt.carbon_3.0.0/os/macosx/ppc

Для створення профіля налаштування програми потрібно викликати меню «Run → Debug». Далі додати нову конфігурацію налагодження «Java Application» і вказати необхідні аргументи для Java машини (рис. 2.3.1) [9].

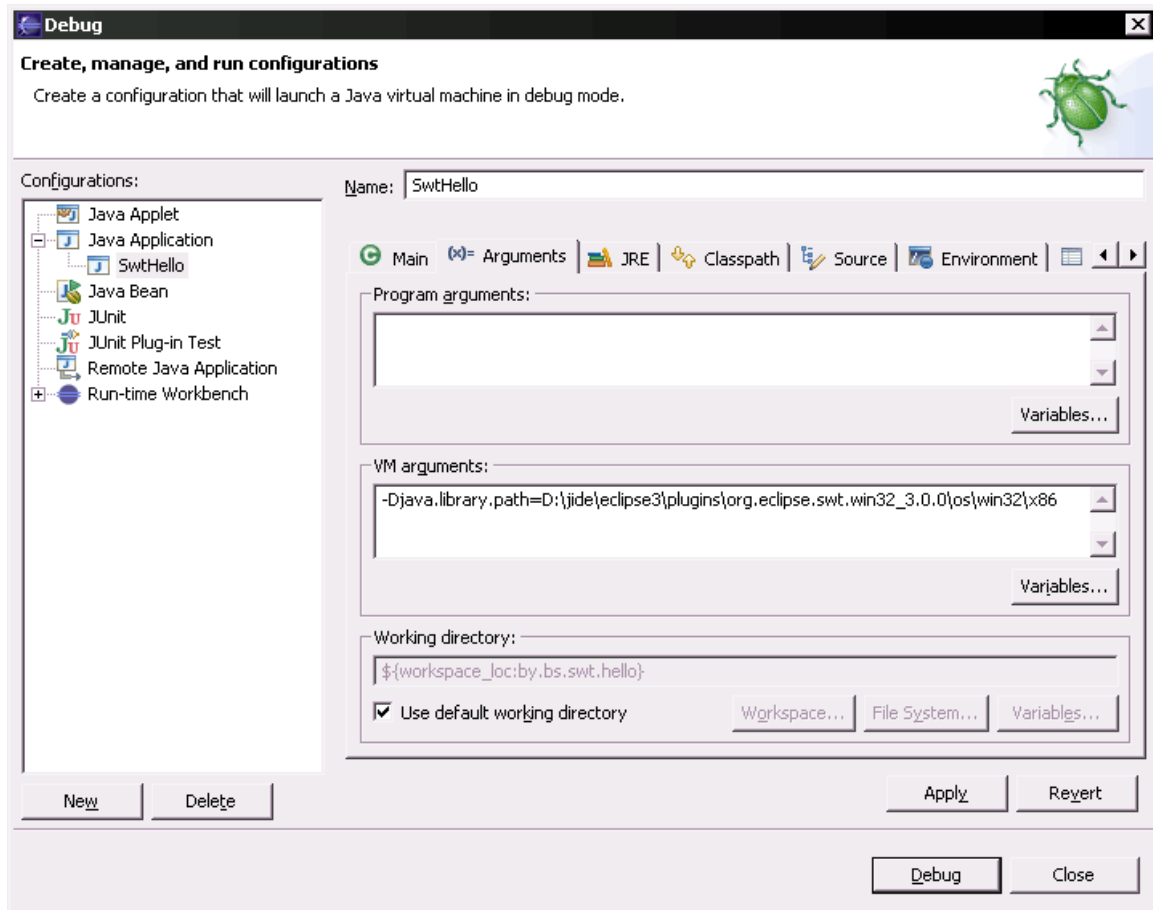


Рис. 2.3.1 – Налаштування параметрів налагодження

Після цього бібліотеку SWT можна підключати до проектів. Для цього слід у формі налаштувань проекту на вкладці «Projects» натиснути кнопку «Add» і в окремому вікні додати до проекту бібліотеку SWT у вигляді додаткового проекту (org.eclipse.swt). Структура проекту з використанням бібліотеки SWT в IDE Eclipse представлена на рисунку 2.3.2.

Об'єкт Display, який використовується в Java-додатках з SWT, в інтерфейсі невидимий, але потрібен для того, щоб відображати графічні візуальні компоненти, розташовані на ньому.

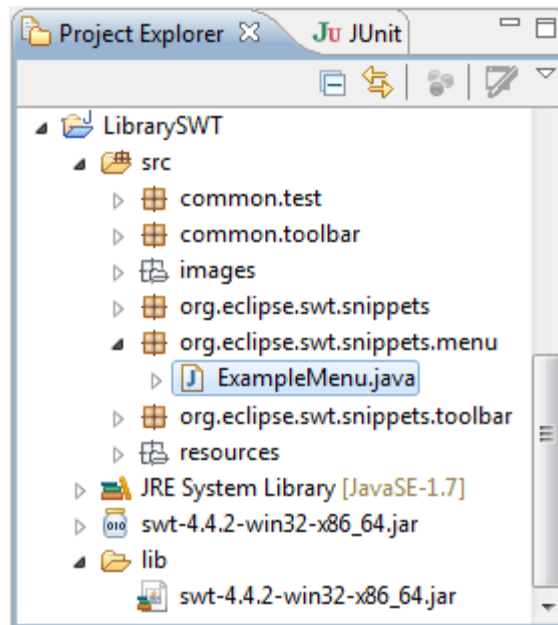


Рис. 2.3.2 – Структура проекту з використанням бібліотеки SWT

Об'єкт Shell – це вікно в додатку. Для відображення графічного інтерфейсу додатку необхідно відкрити Shell і створити цикл відслідковує знищення вікна. У додатку може бути створено кілька примірників Shell.

Widget – основний компонент графічного інтерфейсу SWT (на кшталт компоненту в пакеті Java AWT і JComponent в Swing). Widget є абстрактним класом.

Composite (комполит) – це елемент управління, який може містити в собі інші елементи. Аналог контейнеру в пакеті Java AWT і JPanel пакету Swing.

Object (об'єкт) – це батьківський клас інших елементів (які можуть і не бути композитами), наприклад, таких як, список або таблиця. Об'єкт є абстрактним класом [8].

Майже всі графічні інтерфейси Java-додатки з SWT створюються з декількох основних частин. Всі SWT widgets можуть бути знайдені в пакетах org.eclipse.swt.widgets або org.eclipse.swt.custom (деякі плагіни Eclipse також надають кастомізуючі widgets в інших пакетах) [9].

Для розробки програми були використані такі віджети:

- org.eclipse.swt.widgets.Composite;



- org.eclipse.swt.widgets.Display;
- org.eclipse.swt.SWT;
- org.eclipse.swt.layout.GridLayout;
- org.eclipse.swt.widgets.Label;
- org.eclipse.swt.widgets.Shell;
- org.eclipse.swt.layout.GridData;
- org.eclipse.swt.widgets.Button;
- org.eclipse.swt.widgets.Group;
- org.eclipse.swt.custom.ScrolledComposite;
- org.eclipse.swt.layout.RowLayout;
- org.eclipse.swt.layout.FillLayout;
- org.eclipse.swt.widgets.TabFolder;
- org.eclipse.swt.widgets.TabItem.

Бібліотека SWT використовує основні widgets (віджети) операційної системи [10]. Додатково спільно з SWT використовується бібліотека JFace, надаючи Java-додаткам з бібліотекою SWT численні сервіси.

З точки зору програмного додатку widget («віджет») – це елемент графічного інтерфейсу, який має стандартний зовнішній вигляд і виконує стандартні дії. Під віджетом маються на увазі вікно (діалогове, модальне), кнопка (стандартна, радіокнопка, флажкова), список (ієрархічний, що розкривається) і т.д.

JFace – це набір Java-класів, що реалізують найбільш загальні завдання побудови GUI . З точки зору розробки Java-додатки JFace є додатковим програмним шаром над SWT, який реалізує шаблон Model-View-Controller. JFace реалізує наступні можливості:

1. Представлення «Viewer» класів, що відповідають за відображення і реалізують завдання щодо заповнення, сортування, фільтрації.
2. Представлення «Action» класів, що дозволяють визначати «поведінку» окремих віджетів, таких як пункти меню, кнопки, списки і т.д.

3. Уявлення реєстрів, що містять шрифти і зображення.

4. Представлення набору стандартних діалогових вікон і віджетів для взаємодії з користувачем.

Для угруповання компонент інтерфейсу використовуються контейнери (Container). Для створення основного контейнера для додатка найчастіше використовується контейнер JFrame (є ще JWindows і JApplet). Найпростіше успадкувати від JFrame тим самим отримати доступ до безлічі методів, наприклад:

- setBounds (x, y, w, h) - вказує координати верхньої лівої вершини вікна, а також його ширину і висоту;
- setResizable (bool) - вказує, чи можна змінювати розмір вікна;
- setTitle (str) - встановлює назву вікна;
- setVisible (bool) - власне відображає вікно;
- setDefaultCloseOperation (operation) - вказує операцію, яка буде проведена при закритті вікна.

Основні елементи управління:

- JLabel - елемент для відображення фіксованого тексту;
- JTextField - простий edit-box;
- JButton - звичайна кнопка (button);
- JCheckBox - елемент вибору (аналог checkbox);
- JRadioButton - радіо кнопка.

При відображенні елементів управління використовуються спеціальні менеджери - LayoutManager. У всіх LayoutManager'ов є методи для додавання у видалення елементів.

FlowLayout - використовується для послідовного відображення елементів. Якщо елемент не поміщається в конкретну рядок, він відображається в наступному.

GridLayout - відображення елементів у вигляді таблиці з однаковими розмірами вічок.

BorderLayout - використовується при відображенні не більше 5 елементів. Ці елементи розташовуються по краях фрейма і в Центрі: North, South, East, West, Center.

BoxLayout - відображає елементи у вигляді рядка або колонки.

GridBagLayout - дозволяє призначати місце розташування і розмір кожного елемента. Це найскладніший, але і найефективніший вид відображення.

Варто ще звернути увагу на обробку подій. Для цього використовуються так звані Event Listeners.

Код графічного інтерфейсу користувача представлений у додатку Б

Висновок до розділу 2

У ході порівняння сучасних мов програмування за стандартними факторами, було обране середовище Java, оскільки вона є крос-платформеною, має простий синтаксис, високу швидкість розробки програми та постійну підтримку зі сторони розробників. Для створення графічного інтерфейсу використовувалася бібліотека Standard Widget Toolkit, яка має відкриті прості шляхи найпопулярніших операційних систем, а також просте налаштування для роботи в різних умовах.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ТЕСТУ АМТХАУЕРА

#### 3.1 Блок-схема алгоритму

Етап І. Користувач запускає програму з тестом, на екрані з'являються питання з 1 розділу, які по мірі заповнення відповідями змінюються на питання з наступного розділу. Якщо питання, на яке відповідає користувач, з 1-3,5-8 розділів, то правильна відповідь позначається обраним радіобаттоном. Якщо питання, на яке відповідає користувач, з 4 розділу, то користувач записує відповідь в інпут своїми словами, оскільки ці питання мають відкриту відповідь. Якщо питання, на яке відповідає користувач, з 9 розділу, то він записує відповідь у відповідну таблицю, оскільки ці питання також мають відкриту відповідь. Якщо відповіді дані на всі питання, то отримані дані записуються у формат .csv та передаються на налаштований комп'ютер, а тест завершується. Якщо відповіді дані не на всі питання, то шлях користувача повертається до питання «З якого розділу це питання?». Блок-схема першого етапу представлена на рисунку 3.1.1.

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						36
Изм.	Лист	№ докум.	Підпис	Дата		

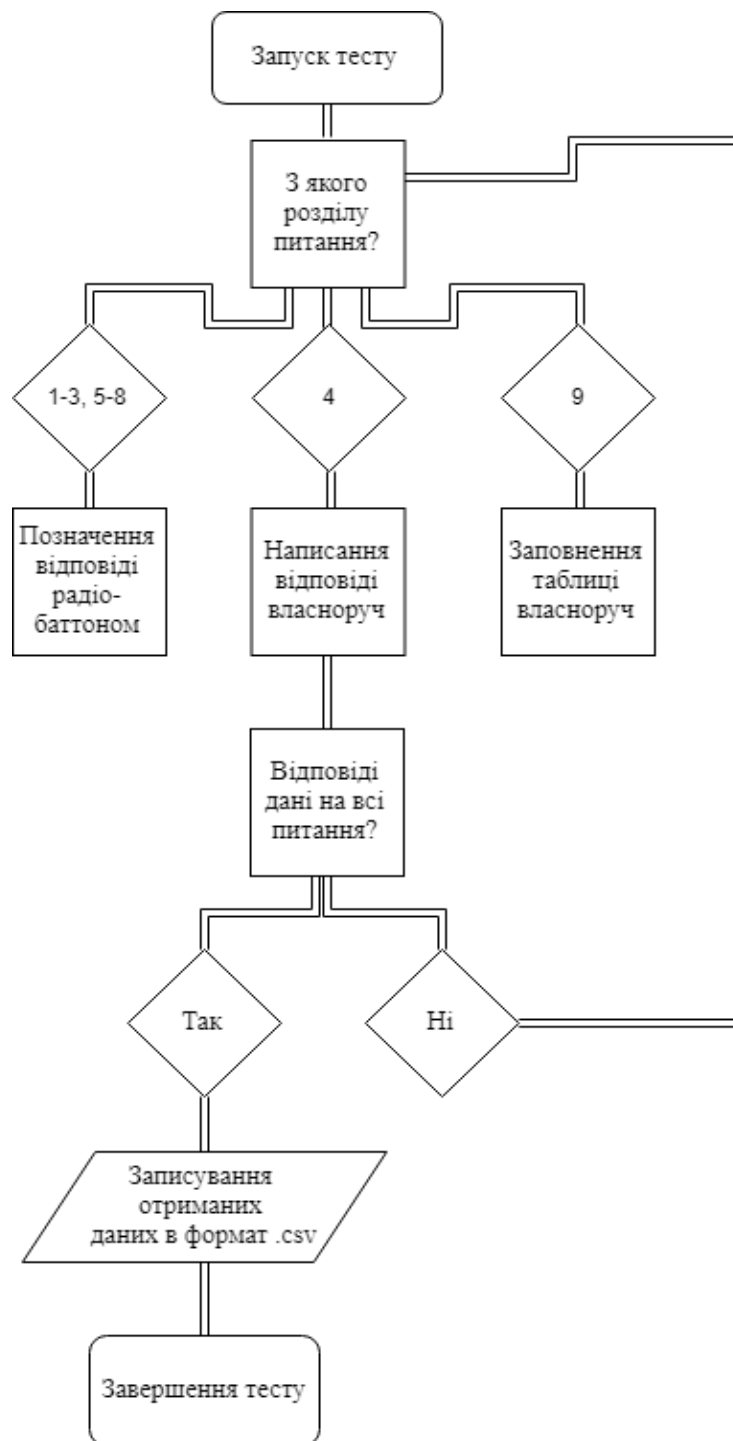


Рис. 3.1.1 – Блок-схема I етапу

Етап II. Після завершення тесту, отримані дані записуються у .csv файл та відкриваються у програмі Excel. Відбувається перевірка на правильність відповідей, тобто перевіряється кожна комірка. Якщо комірка з відповіддю відноситься до 1-3, 5-8 розділів, то за допомогою формули розраховується

результат, тобто якщо відповідь вірна, то додається відповідна кількість балів, якщо відповідь невірна – додається 0. Якщо перевірені всі комірки з 1-3, 5-8 розділів, то підраховується кількість балів і перевірка завершується. Якщо перевірені не всі комірки з 1-3, 5-8 розділів, то повторюється дія з перевіркою результатів у ще не перевірених комірках. Якщо комірка з відповіддю відноситься до 4 розділу та 9 розділу, то користувач власноруч аналізує і перевіряє відповіді на правильність. Блок-схема другого етапу представлена на рисунку 3.1.2.

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
Изм.	Лист	№ докум.	Підпис	Дата		38

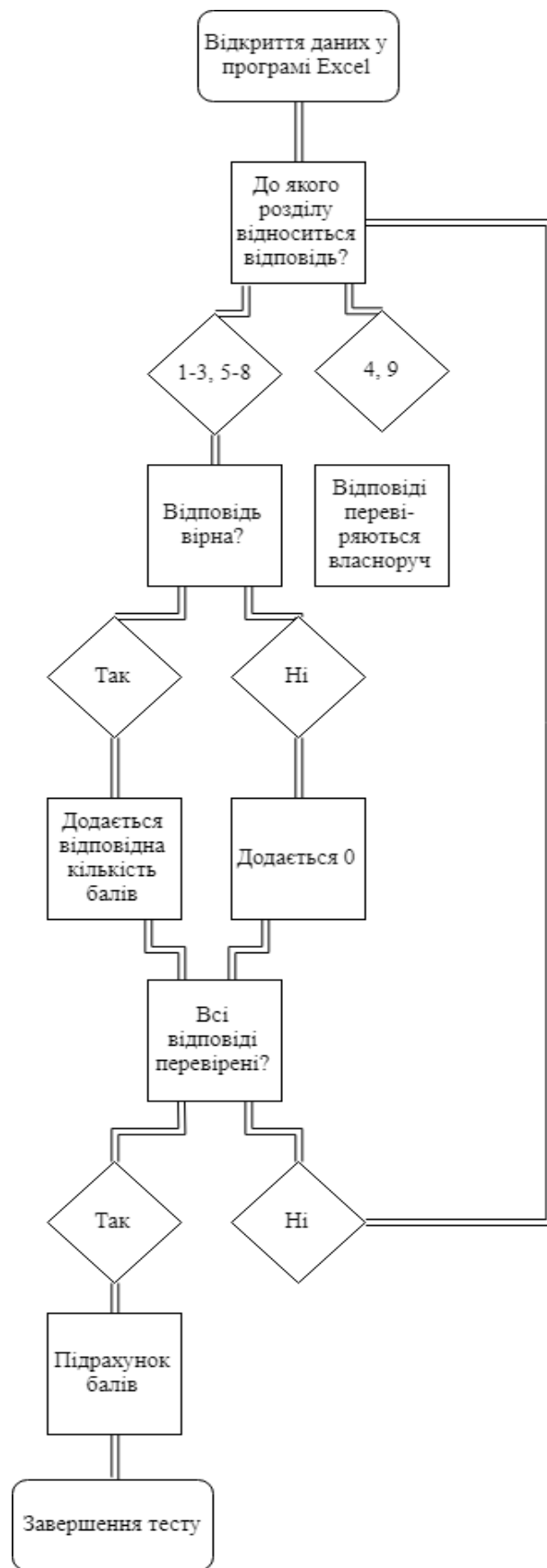


Рис. 3.1.2 – Блок-схема II етапу

### 3.2 Результат програмування

При відкритті програми з'являється дефолтне вікно (рис. 3.2.1) із запитанням готовності користувача до проходження тесту.

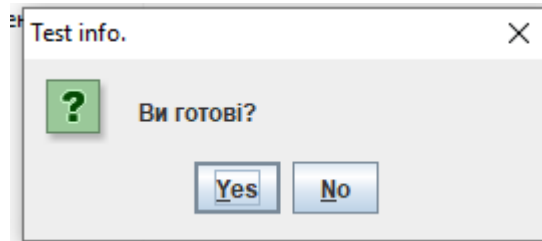


Рис. 3.2.1 – Дефолтне вікно програми

На рисунку 3.2.2 можна побачити результат розроблення одного з вікон програмованого тесту Амтхауера. В питаннях, де є вибір відповіді, може бути тільки одна вірна. Тобто, щоб обрати відповідь на якесь із питань, треба поставити відмітку в одному із радіобатонів. Якщо користувач захоче вибрати додатково другу відповідь у питанні, мітка в радіобатоні буде тільки на тому питанні, що був відмічений останнім. Частково код програми представлений у додатку А.



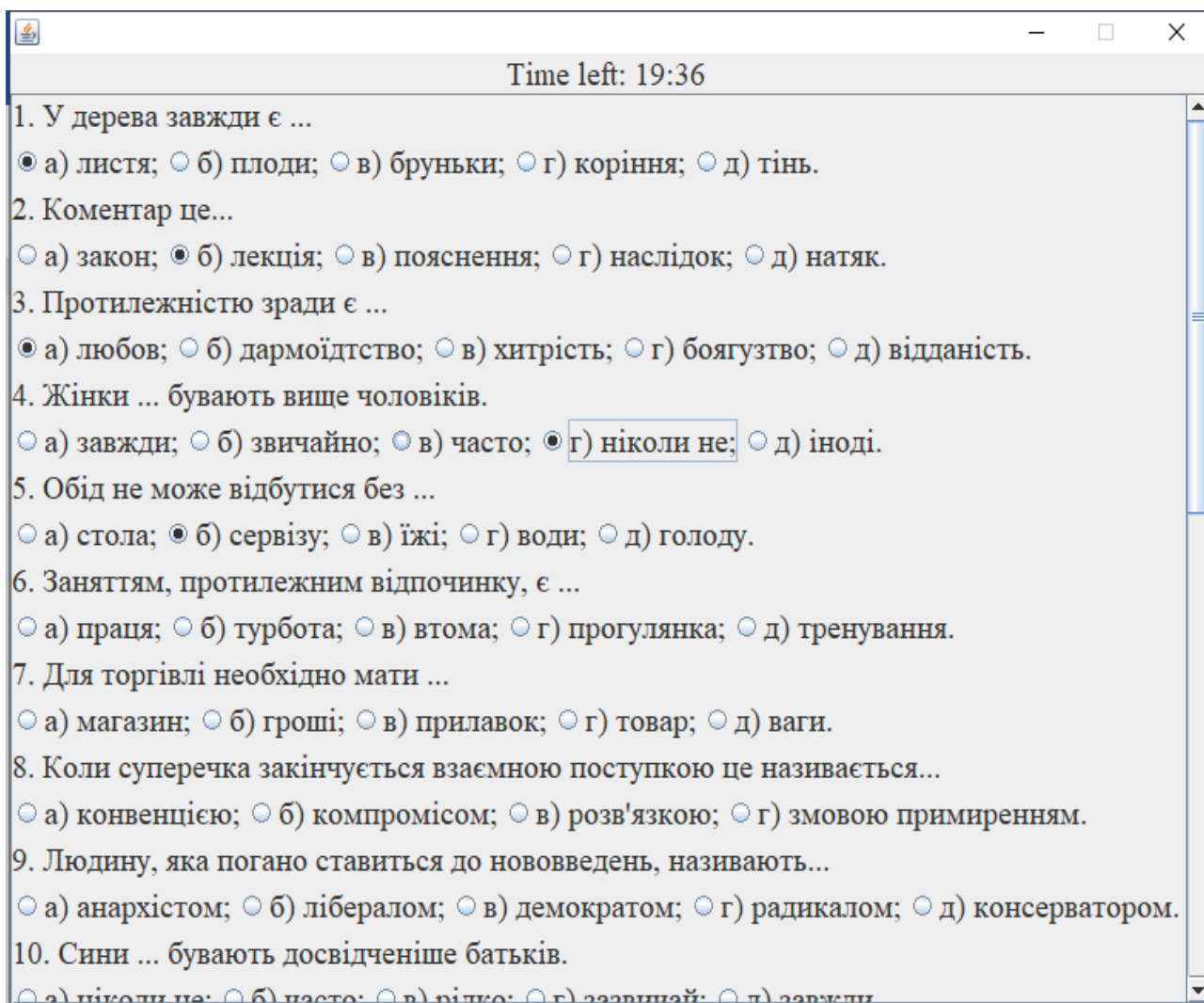


Рис. 3.2.2 – Вікно створеної програми із запитаннями, що мають закриту відповідь

Також, у тесті реалізована можливість вводу власної відповіді користувачем у відповідне поле вооду (рис. 3.2.3)

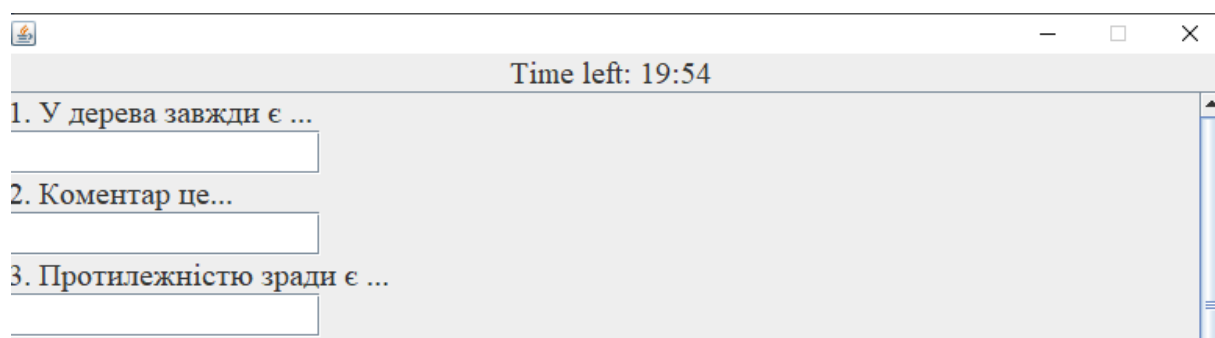


Рис. 3.2.3 – Вікно створеної програми із запитаннями, що мають відкриту відповідь

Після завершення тесту всі введені відповіді записуються у створений програмою файл формату .csv. Якщо відповідь закритого типу – записується буква обраної відповіді, якщо відповідь відкритого типу – записується введена користувачем інформація, якщо відповідь не була надана – ставиться тільда. Скріншот файлу із записаними введеними даними показаний на рисунку 3.2.4.

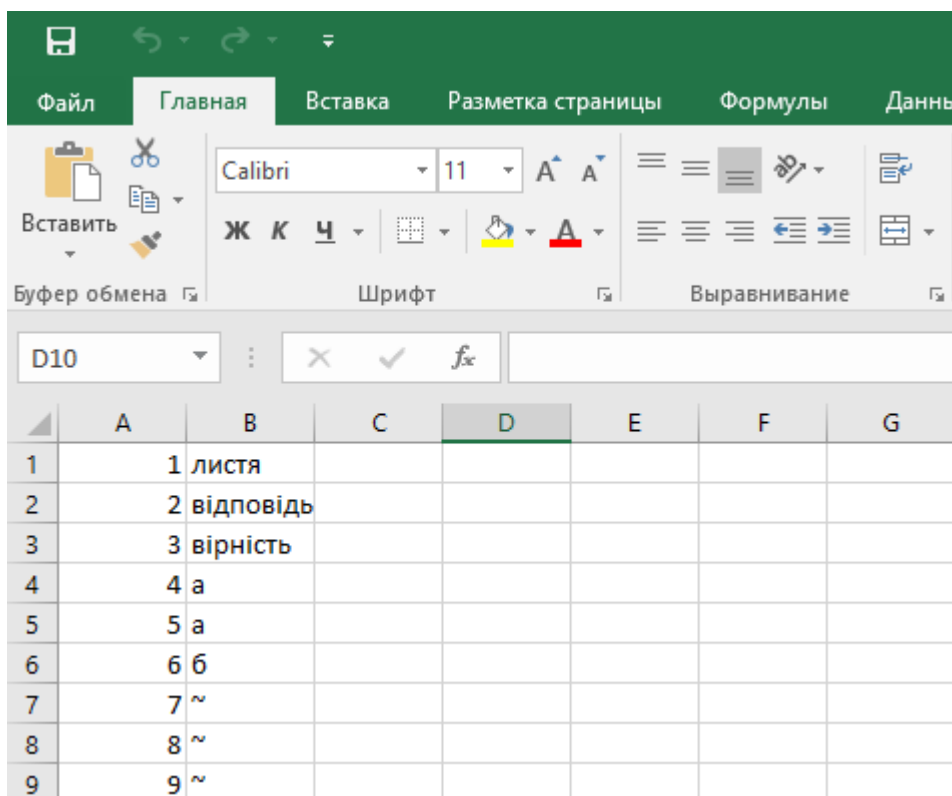


Рис. 3.2.4 – Файл з введеними даними

Оскільки мова програмування даної програми Java, то програма піддається користувацьким змінам налаштування інтерфейсу та шляхів до відповідних файлів. На рисунку 3.2.5 представлено вікно налаштувань програми, які користувач може змінити.

```

properties — Блокнот
Файл Правка Формат Вид Справка
# Max window width & height
800 600

# Path to test data folder
data

# Initialization file extension
.ini
# List of test folders filename
testList
# Name of file with description to such test
description
# Name of file with test duration
duration
# Name of file with test items list
itemList

# Text content files expansion
.txt
# Image content files expansion
.png
# Text content item filename
content
# Item variants list filename (optional)
variants
# Image content item filename
image

# Path to save test results
results
# Test results file extension
.csv
# Test results filename
results

```

Рис. 3.2.5 – Вікно налаштувань програми

### Висновок до розділу 3

У ході виконання практичної частини було створено інтерфейс програми, питання з розділів були розбиті на декілька блоків по 5 питань задля більш зручного використання. Після завершення тесту усі питання будуть передаватися на один комп'ютер в один файл у форматі .csv. При необхідності цей файл можна відкрити в програмі Ехсе І та проаналізувати отримані дані кожного студента.

## РОЗДІЛ 4

### ОХОРОНА ПРАЦІ

Завдання з дипломної роботи має наступну тему: «Програмний комплекс для визначення професії на основі тесту Амтхауера», метою якого розробки програми – тесту, який буде вхідні дані передавати в один файл задля їх перевірки.

#### 4.1. Загальна характеристика умов праці у навчальну приміщені.

Навчальний кабінет, що розглядається, знаходиться на вулиці Янгеля 15/2 та має номер 4-23. Кабінет являє собою світле, сухе приміщення зі світлою побілкою та вкритою лінолеумом підлогою. Вентиляція природня. Освітлення: комбінація штучного та природнього. На рис. 4.1 зображено план приміщення. Параметри навчального кабінету, а також перелік предметів і обладнання наведено в таблиці 4.1.1.

Таблиця 4.1.1 – Характеристики приміщення

№	Найменування	Основні характеристики	Кількість	Позиція на рисунку
Приміщення				
1	Параметри приміщення	5000×7000×2700; S=35 м <sup>2</sup> ; V=94,5 м <sup>3</sup>	–	–
2	Кількість працюючих	1 старший викладач, 14 студентів	15	–
3	Природне освітлення	VEKA WHS 60 1500x700 мм	4	1
4	Штучне освітлення	Світильник ЛПП-158	6	–
5	Вентиляція	Природня	1	-
Обладнання та оснащення				
6	Парта Stod 2-х м. аудиторна без полиці	Розміри: 1200×500×760 Матеріал: бук	7	2

7	Лавка двомісна навчальних закладів Stod для	Розміри: 1000×300×460 Матеріал: сидіння – бук, каркас - метал	7	3
8	Монітор 23.8" Lenovo ThinkVision T2424p	Розміри: 540×140×510 Споживана потужність: 23-32 Вт	14	4
9	Системний блок Dell 780 ATX/2 ядра Intel/DDR3	Розміри: 480x250x520	14	5
10	Клавіатура ACER 5410T	Розміри: 450x120x70	14	6
11	Комп'ютерна миш ACER	Розміри: 185x70x55	14	7
12	Вогнегасник ВП-10	Розміри: 300x300x500 мм Порошковий вогнегасник	1	8

На рис. 4.1.1 зображена схема навчального приміщення

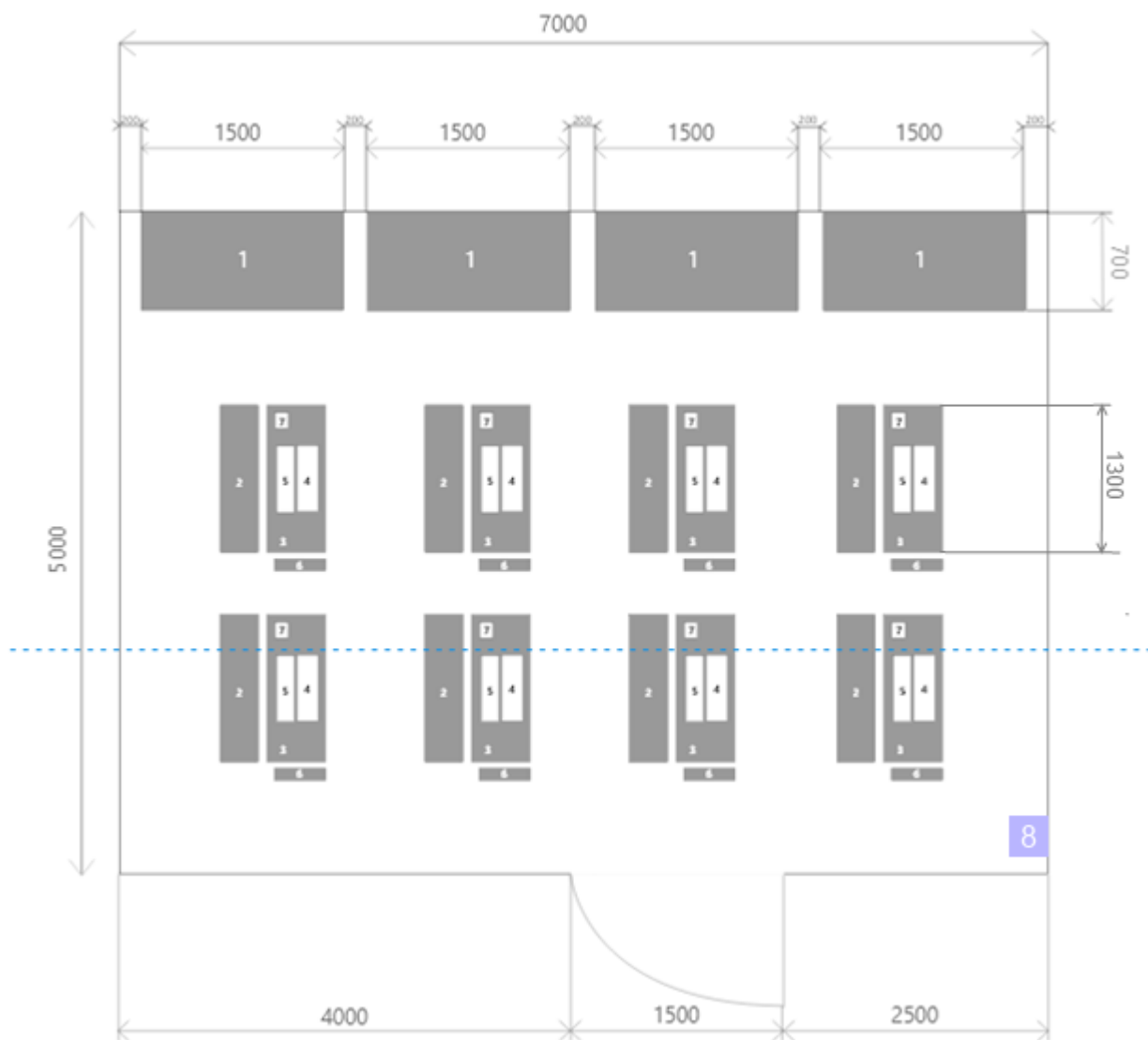


Рисунок 4.1.1 – План навчального приміщення

#### 4.2 Відповідність вимогам об'єму і площі на одного працівника та розташування технологічного обладнання

Таблиця 4.2.1 – Порівняння отриманих даних з нормативними значеннями

Параметри	Нормативне значення	Фактичні значення
Корисна площа на 1 працівника, (м <sup>2</sup> )	4,5	2,5
Корисний об'єм на 1 працівника, (м <sup>3</sup> )	15	6,75
Мінімальна ширина проходу	1.5	1.5

Порівнявши фактичні та нормативні параметри навчальних приміщень (таблиця 4.2.1), можна зробити висновок, що приміщення відповідає всім вимогам. Заходів нормалізації не потрібно.

#### 4.3 Оцінка небезпечних і шкідливих виробничих факторів

Аналіз потенційних небезпек, які створюються на робочих місцях у приміщенні в процесі експлуатації програмно-апаратного комплексу та заходів по покращенню (нормалізації) умов праці наведено в таблиці 4.3.1.

Небезпечні та шкідливі виробничі чинники за їх природою можна поділити на 4 групи, у відповідності до ГОСТу 12.0.003-74. З таблиці 4.3.1 можна побачити їх перелік або відсутність.

Таблиця 4.3.1. – Небезпечні та шкідливі виробничі фактори

Фізичні	Хімічні	Біологічні	Психофізіологічні
Електробезпека, шум, пожежна безпека, освітленість	Відсутні	Відсутні	Відсутні

#### 4.4 Шум

В таблицях 4.4.1 - 4.4.3 наведені джерела та наслідки шуму, а також приведені реальні та нормативні їх значення.

Таблиця 4.4.1.– Основні джерела шуму, що наявні під час проходження тесту

№	Джерело небезпеки	Причини небезпеки	Наслідки небезпеки
1.	Шум з вулиці	зниження концентрації	неефективність тесту
2.	Шум від людської мови у кабінеті	Зниження концентрації	Неефективність тесту

Таблиця 4.4.2 – Реальні та нормативні значення джерел біонебезпеки

№	Фактор небезпеки	Реальне значення	Нормативні значення
1.	Шум	40-70	55

Таблиця 4.4.3. – Заходи та засоби безпеки

№	Група номенклатурних заходів з ОП	Вид заходу	Критерій вибору
1.	Технічні заходи	не передбачені	
2.	Організаційні заходи	інструктаж з питань поведінки	навчання з питань поведінки під час проходження тесту
3.	Режимні	не передбачені	
4.	Експлуатаційні	не передбачені	
5.	ЗІЗ	не передбачені	

#### 4.5 Електронебезпека

Оскільки в приміщенні, де проводиться тест, розміщено прилади, які споживають електроенергію, згідно з нормативами вона відноситься до категорії з підвищеною небезпекою електротравматизму. Джерела небезпеки наведені, реальні та нормативні значення, а також засоби і заходи захисту від ураження електричним струмом наведені в таблицях 4.5.1- 4.5.3.



Таблиця 4.5.1 – Джерела електронебезпеки

№	Найменування обладнання	Джерело небезпеки	Причини небезпеки	Наслідки небезпеки
1.	Монітор 23.8" Lenovo ThinkVision T2424p	мережевий кабель	пошкодження ізоляції (людський та технологічний фактор)	ураження струмом, порушення серцевої діяльності та дихання, отримання інших електротравм
2.	Системний блок Dell 780 ATX/2 ядра Intel/DDR3	мережевий кабель	пошкодження ізоляції (людський та технологічний фактор)	ураження струмом, порушення серцевої діяльності та дихання, отримання інших електротравм

Таблиця 4.5.2 – Реальні та нормативні значення для струму

№	Фактор небезпеки	Реальне значення	Нормативні значення
1.	Змінний струм в мережевих кабелях живлення.	45-60 А	60 мА змінного струму

Таблиця 4.5.3 – Заходи та засоби захисту від ураження електричного струму

№	Група номенклатурних заходів з ОП	Вид заходу	Критерій вибору
1.	Технічні заходи	ізоляція підлоги лінолеумом	електроізоляція
		своєчасна заміна деталей техніки, що вийшли з ладу чи пошкоджені	контроль працездатності приладів
2.	Організаційні заходи	інструктаж з питань електробезпеки	навчання з питань безпеки при експлуатації обладнання
3.	Режимні	не передбачені	
4.	Експлуатаційні	перевірка комп'ютеру на відсутність вірусів	доцільність використання
5.	ЗІЗ	непередбачені.	

Дане приміщення класифікується, як без підвищеної небезпеки, впродовж доби температура в навчальному приміщенні не перевищує 29°C; відносна вологість менше 75. Підлога не є струмопровідною та в приміщенні відсутній струмопровідний пил, що може бути додатковим джерелом електронебезпеки.

#### 4.6 Пожежна безпека

Таблиця 4.6.1.– Основні джерела пожежної небезпеки, які створюються при експлуатації великої кількості комп'ютерів

№	Найменування обладнання	Джерело небезпеки	Причини небезпеки	Наслідки небезпеки
1.	Монітор 23.8" Lenovo ThinkVision T2424p	мережевий кабель	пошкодження ізоляції (людський та технологічний фактор)	виникнення пожежі, що може завдати шкоди людині, а також знищити цінне обладнання.
2.	Системний блок Dell 780 ATX/2 ядра Intel/DDR3	мережевий кабель	пошкодження ізоляції (людський та технологічний фактор)	виникнення пожежі, що може завдати шкоди людині, а також знищити цінне обладнання.

Таблиця 4.6.2 – Характеристика приміщення

Класи та підкласи можливих пожеж	клас А (А1)	Горіння твердих речовин, що супроводжується тлінням
	клас Е	Горіння електроустановок під напругою
Групи горючості матеріалів і речовин, які присутні у даному приміщенні	Негорючі (неспалимі) та горючі (спалимі). Речовини, які здатні до самозагоряння у даному приміщенні відсутні.	
Категорія пожежної небезпеки приміщення	категорія Д, зона класу П-Па	Простір у приміщенні, у якому знаходяться тверді горючі речовини та матеріали.

Таблиця 4.6.3. – Заходи та засоби захисту від пожежі

№	Група номенклатурних заходів з ОП	Вид заходу	Критерій вибору
1.	Технічні заходи	Автоматичні засоби гасіння пожеж та сигналізації, що забезпечують сповіщення про початок пожежі	індикація виникнення пожежі
		Використання вогнегасника ВП-10	спосіб гасіння пожежі
		Плановий інструктаж з техніки безпеки та евакуації	навчання з питань пожежної безпеки при експлуатації обладнання та евакуації
2.	Організаційні заходи	Плановий огляд комп'ютерної техніки, вчасне виявлення і усування несправностей	контроль обладнання
		Безпечне розташування елементів електронних схем один від одного (дроти, кабелі)	
3.	Режимні	не передбачені	
4.	Експлуатаційні	перевірка комп'ютеру на відсутність вірусів	Достовірність інформації
5.	ЗІЗ	протигази, респіратори та маски, ізолюючі саморятівники	індивідуальний захист

## Висновок до розділу 4

Було розглянуто та оцінено потенційні небезпечні та шкідливі фактори, що можуть виникнути при реалізації програмного комплексу. У даному приміщенні значення шуму та освітлення не перевищують нормативних значень. Електробезпека була зведена до мінімуму. При аналізі пожежної безпеки було визначено, що в приміщенні проводяться всі заходи пожежної безпеки.

## ВИСНОВКИ

В ході виконання даної роботи було ознайомлено із структурою тесту структури інтелекту Амтхауера та проаналізовано список необхідної літератури.

Розглянуто середовища програмування та проведено їх порівняння за найвпливовішими факторами. Для реалізації тесту в навчальних цілях обрано мову програмування Java.

Розроблено програму, яка є комп'ютерним додатком – тестом. У програмі реалізовано виведення питань у вигляді тексту та зображень, а також можливість введення відповіді у текстове поле і вибір варіанту відповіді у вигляді радіобаттону.

Для оптимізації навчального процесу, була реалізована передача введених даних з усіх комп'ютерів в один файл задля перевірки відповідей на правильність і аналіз отриманих даних спеціалізованим фахівцем.

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						52
Изм.	Лист	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тест структуры интеллекта Амтхауэра [Электронный ресурс]: <http://childpsy.ru> – Режим доступа: <http://childpsy.ru/lib/metodics/15378/>– Дата доступа: 13.04.2019
2. Тест структуры интеллекта Амтхауэра [Электронный ресурс]: <https://psylab.com.ua> - Режим доступа: [https://psylab.com.ua/tools\\_Тест\\_Структуры\\_Интеллекта\\_Амтхауэра](https://psylab.com.ua/tools_Тест_Структуры_Интеллекта_Амтхауэра) - Дата доступа: 20.04.2019
3. Кинякина О. Н. Мозг на 100%. Интеллект. Память. Креатив. Интуиция / Кинякина О.Н. //— М. : Эксмо. - 2014. — С. 56-57.
4. Еникеев М.Н. Общая психология. Учебник для ВУЗов. / Еникеев М.Н. // - М.: Приор, - 2000 – С. 14-26
5. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. // - М.: Вильямс - 2015. – С. 564-570
6. Гарнаев, Андрей WEB-программирование на Java и JavaScript / Андрей Гарнаев , Сергей Гарнаев. // - Москва: СПб. [и др.] : Питер - 2017. – С. 145-289
7. Гонсалвес, Энтони Изучаем Java EE 7 / Энтони Гонсалвес //. - М.: Питер - 2016. – С. 50-68
8. Савитч, Уолтер Язык Java. Курс программирования / Уолтер Савитч. // - М.: Вильямс - 2015. – С. 465-473
9. Хабибуллин, Ильдар Самоучитель Java / Ильдар Хабибуллин. //- М.: БХВ-Петербург - 2014. – С. 768
10. Шилдт, Герберт Java 8. Руководство для начинающих / Герберт Шилдт. // - М.: Вильямс - 2015. – С. 412-416

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
Изм.	Лист	№ докум.	Підпис	Дата		53

11. Эккель, Брюс Философия Java / Брюс Эккель.// - М.: Питер - 2016. – С. 89
12. Fabio, Luigi Bellifemine Developing Multi-Agent Systems with JADE / Fabio Luigi Bellifemine.// - Москва: ИЛ - 2016. – С. 150-178
13. Васильев, А. Н. Самоучитель Java с примерами и программами / А.Н. Васильев.// - М.: Наука и техника - 2016. – С. 54-56
14. Герберт, Шилдт Java 8. Руководство для начинающих / Шилдт Герберт. // - М.: Диалектика / Вильямс - 2015. – С. 675
15. Сеттер, Р. В. Изучаем Java на примерах и задачах / Р.В. Сеттер. // - М.: Наука и техника - 2016. – С. 140
16. Java Руководство для начинающих [Электронный ресурс]: <https://www.rulit.me> – Режим доступа: <https://www.rulit.me/books/java-rukovodstvo-dlya-nachinayushchih-lp-read-386440-1.html> - Дата доступа: 13.04.2019
17. Ренеган Э.Дж.(мл.) 1001 адрес WEB для программистов: Новейший путеводитель программиста по ресурсам World Wide Web:Пер.с англ./ Ренеган Э.Дж. (мл.).// -Минск:Попурри – 1997.- С.515
18. Чен М.С. и др. Программирование на JAVA:1001 совет: Наиболее полное руководство по Java и Visual J++ :Пер.с англ./Чен М.С.,Грифис С.В.,Изи Э.Ф.// -Минск:Попурри - 1997. – С.512
19. SWT: The Standard Widget Toolkit [Электронный ресурс]: <https://www.eclipse.org> – Режим доступа: <https://www.eclipse.org/swt/> – Дата доступа: 13.04.2019
20. Пакеты SWT и JFace Часть 1 [Электронный ресурс]: <https://www.ibm.com> – Режим доступа: <https://www.ibm.com/developerworks/ru/library/os-jface1/index.html>– Дата доступа: 13.04.2019
21. Eclipse IDE for Java Developers [Электронный ресурс]: <https://www.eclipse.org> – Режим доступа:

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
Изм.	Лист	№ докум.	Підпис	Дата		54

<https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers>– Дата доступа: 19.05.2019

22. Первая программа в Eclipse [Электронный ресурс]: <https://metanit.com> – Режим доступа: <https://metanit.com/java/tutorial/1.4.php>– Дата доступа: 19.05.2019

23. Как запустить Eclipse [Электронный ресурс]: <http://qaru.site> – Режим доступа: <http://qaru.site/questions/676987/how-to-run-a-simple-java-program-in-eclipse>– Дата доступа: 19.05.2019

24. AWS Toolkit for Eclipse [Электронный ресурс]: <https://aws.amazon.com> – Режим доступа: <https://aws.amazon.com/ru/eclipse/>– Дата доступа: 19.05.2019

25. Using the Eclipse IDE For Java Programming [Электронный ресурс]: <https://www.vogella.com> – Режим доступа: <https://www.vogella.com/tutorials/Eclipse/article.html>– Дата доступа: 19.05.2019

26. Configuring Eclipse with SAP Cloud Platform Tools for Java [Электронный ресурс]: <https://developers.sap.com> – Режим доступа: <https://developers.sap.com/finland/tutorials/hcp-java-eclipse-setup.html>– Дата доступа: 19.05.2019

27. How to Configure Webdriver-Selenium for Java in Eclipse on Windows [Электронный ресурс]: <https://saucelabs.com> – Режим доступа: <https://saucelabs.com/resources/articles/how-to-configure-webdriver-selenium-for-java-in-eclipse-on-windows>– Дата доступа: 21.05.2019

28. Java Руководство для начинающих [Электронный ресурс]: <https://www.rulit.me> – Режим доступа: <https://www.rulit.me/books/java-rukovodstvo-dlya-nachinayushchih-lp-read-386440-1.html> - Дата доступа: 06.06.2019

29. Чен М.С. и др. Программирование на JAVA:1001 совет: Наиболее полное руководство по Java и Visual J++ :Пер.с англ./Чен М.С.,Грифис С.В.,Изи Э.Ф.//–Минск:Попурри - 1997. – С.512

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
Изм.	Лист	№ докум.	Підпис	Дата		55

30. Пакеты SWT и JFace Часть 1 [Электронный ресурс]:  
<https://www.ibm.com> — Режим доступа:  
<https://www.ibm.com/developerworks/ru/library/os-jface1/index.html>— Дата  
 доступа: 06.06.2019

					<b>БМ51.08.2705 .1404.ПЗ</b>	Лис
						56
Изм.	Лист	№ докум.	Підпис	Дата		



## ДОДАТОК А

### Лістинг коду.

[illegible]

					БМ51.08.2705 .1404.ПЗ	Лис
Изм.	Лист	№ докум.	Підпис	Дата		57



```

122.
123.     Button button_19 = new Button(composite_4, SWT.RADIO);
124.     button_19.setText(«Radio Button»);
125.
126.     TabItem tabItem_4 = new TabItem(tabFolder, SWT.NONE);
127.     tabItem_4.setText(«6-10»);
128.
129.     Composite composite_6 = new Composite(tabFolder, SWT.NONE);
130.     tabItem_4.setControl(composite_6);
131.     composite_6.setLayout(new GridLayout(1, false));
132.
133.     Composite composite_7 = new Composite(composite_6, SWT.NONE);
134.     composite_7.setLayoutData(new GridData(SWT.FILL, SWT.FILL, false, false, 1, 1));
135.     composite_7.setLayout(new GridLayout(5, false));
136.
137.     Label label_4 = new Label(composite_7, SWT.NONE);
138.     label_4.setLayoutData(new GridData(SWT.LEFT, SWT.CENTER, false, false, 5, 1));
139.     label_4.setText(«New Label»);
140.
141.     Button button_20 = new Button(composite_7, SWT.RADIO);
142.     button_20.setText(«Radio Button»);
143.
144.     Button button_21 = new Button(composite_7, SWT.RADIO);
145.     button_21.setText(«Radio Button»);
146.
147.     Button button_22 = new Button(composite_7, SWT.RADIO);
148.     button_22.setText(«Radio Button»);
149.
150.     Button button_23 = new Button(composite_7, SWT.RADIO);
151.     button_23.setText(«Radio Button»);
152.
153.     Button button_24 = new Button(composite_7, SWT.RADIO);
154.     button_24.setText(«Radio Button»);
155.
156.     Composite composite_8 = new Composite(composite_6, SWT.NONE);
157.     composite_8.setLayout(new GridLayout(5, false));
158.
159.     Label label_5 = new Label(composite_8, SWT.NONE);
160.     label_5.setLayoutData(new GridData(SWT.LEFT, SWT.CENTER, false, false, 5, 1));
161.     label_5.setText(«New Label»);
162.
163.     Button button_25 = new Button(composite_8, SWT.RADIO);
164.     button_25.setText(«Radio Button»);
165.
166.     Button button_26 = new Button(composite_8, SWT.RADIO);
167.     button_26.setText(«Radio Button»);
168.
169.     Button button_27 = new Button(composite_8, SWT.RADIO);
170.     button_27.setText(«Radio Button»);
171.
172.     Button button_28 = new Button(composite_8, SWT.RADIO);
173.     button_28.setText(«Radio Button»);
174.
175.     Button button_29 = new Button(composite_8, SWT.RADIO);
176.     button_29.setText(«Radio Button»);
177.
178.     Composite composite_9 = new Composite(composite_6, SWT.NONE);
179.     composite_9.setLayout(new GridLayout(5, false));
180.
181.     Label label_6 = new Label(composite_9, SWT.NONE);
182.     label_6.setLayoutData(new GridData(SWT.LEFT, SWT.CENTER, false, false, 5, 1));
183.     label_6.setText(«New Label»);
184.
185.     Button button_30 = new Button(composite_9, SWT.RADIO);
186.     button_30.setText(«Radio Button»);

```

## ДОДАТОК Б

### Лістинг коду

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class SimpleGUI extends JFrame {
    private JButton button = new JButton("Press");
    private JTextField input = new JTextField("", 5);
    private JLabel label = new JLabel("Input:");
    private JRadioButton radio1 = new JRadioButton("Select this");
    private JRadioButton radio2 = new JRadioButton("Select that");
    private JCheckBox check = new JCheckBox("Check", false);

    public SimpleGUI() {
        super("Simple Example");
        this.setBounds(100, 100, 250, 100);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container container = this.getContentPane();
        container.setLayout(new GridLayout(3, 2, 2, 2));
        container.add(label);
        container.add(input);

        ButtonGroup group = new ButtonGroup();
        group.add(radio1);
        group.add(radio2);
        container.add(radio1);

        radio1.setSelected(true);
        container.add(radio2);
        container.add(check);
        button.addActionListener(new ButtonEventListener());
        container.add(button);
    }

    class ButtonEventListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            String message = "";
            message += "Button was pressed\n";
            message += "Text is " + input.getText() + "\n";
            message += (radio1.isSelected()?"Radio #1":"Radio #2")
                + " is selected\n";
            message += "CheckBox is " + ((check.isSelected())
                ?"checked":"unchecked");
            JOptionPane.showMessageDialog(null,
                message,
                "Output",
                JOptionPane.PLAIN_MESSAGE);
        }
    }

    public static void main(String[] args) {
        SimpleGUI app = new SimpleGUI();
        app.setVisible(true);
    }
}
```